## CONTENTS:


1. Introduction
2. Installation Set-Up
2. Limitations
3. System Installation Drawing
4. Getting Started
5. The Directory
5. Disc System Commands
6. The Command Extensions
7. Entering Commands
9. Specification of Disc Commands By Variables
9. To Incorporate Disc Commands into Tape Programs
1Ø. How to Erase
1Ø. Selecting Drives
11. Disc Capacity
11. How to Make Back-Ups
11. How to Auto Boot Your Programs
12. System Upgrades and Changes
13. Expanding the on-board RAM to 256K Bytes
13 Source Code Availability
14. Troubleshooting Problems
16. Disc Operation Theory
17. FD-68 Theory of Operation
18. Data Bus Diagram
18. Address Bus Diagram
19. Control Logic Schematic
2Ø. Disc Controller Schematic
21. Dynamic RAM Details
21 RGB Video Details
22. Assembly Drawing
23. Disc Drive Notes

# INTRODUCTION:

Welcome to the **AERCO FD-68 Floppy Disc Interface**. This interface is designed to control up to **4** Shugart compatible disc drives. Size can be **3 inch** to **8 inch**, single or double sided **(SS/DS)**. 8 inch drives are limited to single density **(SD)**. All other drives can be single,double, or quad density **(SD/DD/QD)**. In addition to the interface, the minimum disc system requires at least one disc drive and a power supply with **+5v/2A** and **+12v/2A**. The 34-conductor signal cable is also required to connect the interface and disc drives together.

Some of the features included with the interface are:

64K of additional RAM for a total of **112K.** The additional RAM is located in the **DOCK BANK** and can be accessed in **8** each **8K CHUNKS**. (The Z-8ØA processor is, of course, only capable of addressing 64K at one time). The on-board Ram can be expanded to **256K** if desired.

**8K ROM** to hold the Disc Operating System (DOS).

**RGB Output**. Provides a clear and stable video display. The cable is additional and monitor type must be specified.

**Fully compatible** with disc drives and power supply from TS/1ØØØ.

**No** need to modify the computer.

Disc drives are available from AERCO or you can use your own. Drive parameters can be mixed or matched. You can use any drives you want as long as the size, number of sides, track density (SD/DD/QD) and track to track access time are specified for each drive so we can burn the proper EPROM. Replacement EPROMs customized for any combination of drives are available. There is no additional charge to customize the original EPROM shipped with the interface.
Drives can be wired for power in 2 ways. Separate 3-conductor cables can be run from the power supply to each drive, or , **pins 2** and **34** of the signal cable can be used for **+5v** and **+12v** respectively. If the signal cable is used to carry power, as with the TS/1ØØØ system, the disc drives must be modified as shown in the **SYSTEM INSTALLATION DRAWING.**

CAUTION: CUT ANY TRACES CONNECTED TO PINS 2 OR 34 OF THE DISC DRIVE EDGE CONNECTOR IF YOU MODIFY THE DRIVE FOR POWER THRU THE SIGNAL CABLE.

The external power supply is used for the disc drives only. It has 3 wires on each output, **+5v/+12v/gnd.** The interface is powered by the computer.

## INSTALLATION SET-UP:

**NOTE**
**Refer to SYSTEM INSTALLATION DRAWING**

1. Remove power from the computer and the disc power supply.
2. Install the FD-68 Interface on the rear of the computer.
3. Connect the **single** connector of the **34-conductor signal cable** to the **34-pin edge connector** of the interface.
4. Connect other accessories such as the printer interface to the feed-thru on the back of the disc interface.
5. Connect the RGB cable to the short edge connector of the disc interface.

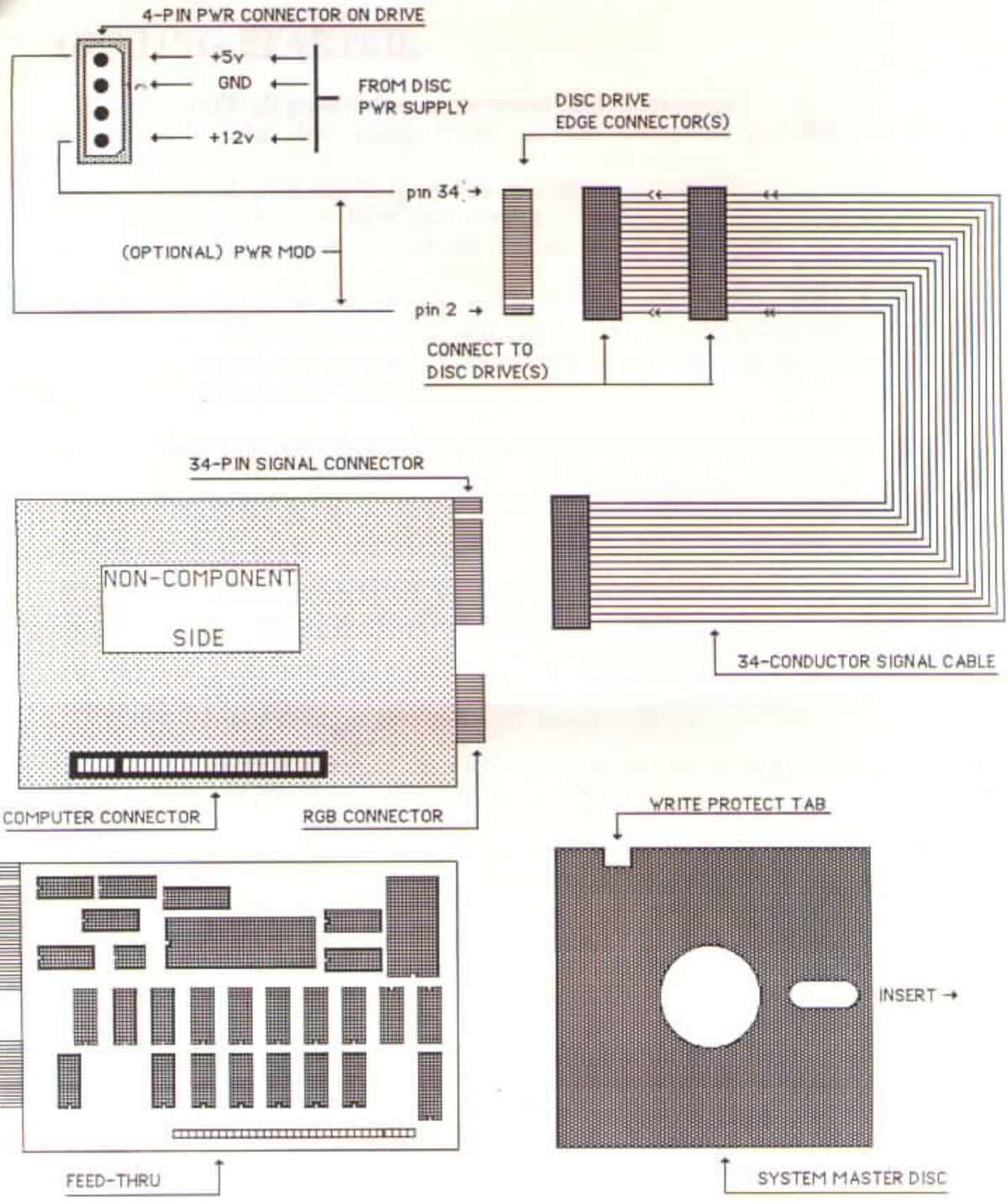
If your drive(s) are not already prepared:

6. Install the connector on the other end of the signal cable to the edge connector of the disc drive.
7. Connect the remaining signal connector to a second drive.
8. Connect the 3-conductor power cable to the disc drive(s).
9. Verify that **Drive Select** settings and **Terminating Resistor** placement are correct on all drives. Refer to **DISC DRIVE NOTES.**

**CAUTION: CHECK ALL CONNECTIONS BEFORE APPLYING POWER.**

## LIMITATIONS as of 8/15/85:

NOTE: The .LRO .CHR and .DAT extensions are not yet implemented. They will be provided on updates soon.




SYSTEM INSTALLATION
DRAWING

## GETTING STARTED:

1. Switch **OFF** all power to your computer and accessories.
2. Assemble the disc components to your computer per the preceeding instructions.
3. Remove shipping inserts from disc drive(s).
4. Insert **System Boot Disc** into drive **A**.
5. Switch **ON** the computer first and then the disc power supply.

The system should **auto boot**. The red busy light will come on drive **A**, the disc will spin and the read/write head will step (click) several times. When the boot program is loaded, the drive will stop and you should see a welcome screen (First .SCR) with our logo and 3 menu selections.

**A for Amazing Demonstration:** is a slick program showing off the system.

**D for SERIAL / PAR. Drivers:** loads the print driver and terminal emulator codes.

**I for Operating INSTRUCTIONS:** provides a few screens of condensed instructions on the esential operations of the disc system.

Although it is not listed, any other key will load the **DIRECTORY** of the disc contents.

Press **i/I** for the first screen of instructions. Press any key except N to step thru the screens. The last step should return to the welcome screen.

Press **a/A** for the demonstration. The video is impressive if you have an RGB monitor. When the program is over, it will load the Directory and you will see a screen with a format similar to the following:

**AERCO - FD68 V.8 DISC A:**
**I.D. T1234 2Ø68 BOOT**

**DIRECTORY 28ØK REMAIN**

| | |
|---|---|
| WOW.BAS | BOOT.BAS |
| DRIVERS.BIN | PORTS.BAS |
| 232 Config.BAS | 232 ReadMe.BAS |
| PRINTYPE.BAS | Timer.BAS |
| Adder.BAS | First.SCR |
| WOW.BIN | YANKEE.BAS |
| ROSE.BIN | ROSE.BAS |
| AFLAG.SCR | TFLAG.SCR |

## THE DIRECTORY:

The directory is a single screen showing all of the files on the disc. All programs are accessed according to their listing in the directory. To access a program, enter it's name exactly as it is listed . It does not matter if you use upper or lower case letters, but the spelling of the name and the extension must be correct.
The maximum number of entries to the directory is **31**. When the directory is full, the system will report the disc is full even though there may be considerable space left on the disc.
The command to view the directory is: **CAT "",**
The directory can be viewed without disturbing system memory. You can usually return to where you were before viewing the directory with the command **GOTO1** or **CONTINUE.**

## DISC SYSTEM COMMANDS:

There are **4** keyboard commands used with the disc system. They are accessed by selecting the **E cursor** and holding **SYMBOL SHIFT** while depressing the desired key.

**CAT** (E/ S-SHIFT 9): Is the equivalent of LOAD. Use it to load programs from the disc into system memory. The desired program must be listed in the disc directory.

**MOVE** (E/ S-SHIFT 6): Is the equivalent of SAVE. Use it to write programs from system memory onto the disc.

**ERASE** (E/ S-SHIFT 7): Is used to remove programs from the disc. Enter the name and extension as it appears in the disc directory. The remaining K on the disc will increase by the size of the program erased and that entry will be removed from the directory listing.

**FORMAT** (E/ S-SHIFT Ø) Is used to record track and sector headers on a new disc. The exact number and size of the headers depend on what parameters are programmed into the **EPROM**. They can be the same or different for all four drives. Format can also be used to completely erase an old disc. The system will warn you with the following message if there is already data on the disc you are going to format:

**DISC A HAS DATA ! OK TO ERASE?**
**Y/N**

Entering any charactor other than **Y/y** will cancel the format. If you try to FORMAT a disc with the Write Protect Tab installed, the following report will appear:

**DISC A WRITE PROTECTED**

## THE COMMAND EXTENSIONS:

The extension defines the type of information in a program: such as BASIC or machine code. The extension is defined by a **period** and **three letters** after the name of your program. They can be entered as upper or lower case. The following is a list of the extensions:

**.ARO** AROS runs in the dock bank starting at 8ØØØ Hex.
**.BAS** BASIC program and its variables.
**.BIN** Binary machine code.
**.BUT** BOOT program to be executed from cold start. (CP/M only)
**.CHR** CHARACTER data.
**.DAT** NUMERIC data.
**.LRO** LROS runs in the dock bank starting at Ø.
**.SCR** SCREEN is a copy of the video display.
**.VAR** All BASIC variables.

**The .ARO extension:** SAVES/LOADS memory in the Dock Bank, starting at address 8ØØØH. The Chunks (8 x 8K) are selected and the program is started per the standard TIMEX rules for operating an AROS (Application ROM Orientated Software). Instead of running it from ROM, you are running it from RAM as loaded from the disc.

**The .BAS extension:** SAVES/LOADS BASIC program and data areas. An **optional Decimal parameter** may be included in the save command to specify the starting Line Number. A different start location can be specified when loaded from disc by an **optional Decimal parameter** included in the command to load.

**The .BIN extension:** SAVES/LOADS binary machine code data. It is the equivalent of CODE in the tape system. **Two Decimal Parameters** are required when saving: the **Start address** and the **number of Bytes**. No parameters are required to LOAD the CODE back into system memory. A machine code program may be loaded into a different memory location by specifying an **optional Decimal parameter** in the load command. The Chunk Specification (details of the Bank Switching configuration) is saved along with the Binary file and is restored to the same state as the file is loaded from disk. This means that you can save on disk Binary Files that reside in other than the Home Bank and reload them from disk directly into the bank from which they originated.

**The .BUT extension:** Is an assembly language program written at 341ØH (D41ØH in the SPECTRUM mode). It is the cold boot loader in CP/M and other non-BASIC environments. If when you format a disk you specify the "C" option when asked what type of disk you are creating, the system will commence executing the assembly code that was located at 341ØH when it was saved as Name.BUT.

**The .SCR extension:** Is a copy of the screen at the time it is saved. Loading the screen will return the image but not the program that was associated with it. It automaticaly includes the **Second Display File** if it is active.

**The .VAR extension**: Allows you to SAVE all of the BASIC Variables. It does not affect the BASIC Program and can be used to manipulate Variables even when no Program is present. It allows the space allocated to Variables to grow or shrink as required when Variables are added or deleted.

## ENTERING COMMANDS:

All of the commands have the same basic format:

**COMMAND "NAME .EXTENSION", DECIMAL PARAMETER(S)**

The type of command is always a **keyword**. The name can be from **1 to 10** charactors long, except " or , (quote or comma) and may be upper or lower case. The extension is always a **period** followed by **3 letters** which can also be upper or lower case. The **open quote** symbol comes after the command type and the **close quote** comes after the extension; followed by a **comma**. Additional Decimal parameters may be added to the end of the command statement. If more than one Decimal parameter is included, each one must be separated by a comma, but no punctuation is required after the last number.

**CAT** is used to load any file that is listed in the directory. The basic command is:
**CAT "name .extension", optional start line.**
For example, to view the welcome screen enter **CAT "first.scr",**

**FORMAT** Before attempting to write to the disc, a fresh disc should be prepared. It is a good idea not to write on the original System Boot disc.

1. Remove the System Boot disc and install a fresh disc without the write protect tab.
2. Enter the command: **FORMAT "",**
3. The busy light should come on **drive A** and it will step (click) across the disc from the outside in. The bottom of the display will print **FORMATTING** with a * for each track. Double density drives normally have **4Ø** tracks per side (**48** Tracks Per Inch) and quad density is **8Ø** tracks per side (**96 TPI**). Because both sides are formatted for each track, single side drives will step twice as fast as double side drives. When the formatting is complete, the disc will step rapidly (buzz) back to the ouside and the following message will appear at the bottom of the screen.

**TYPE OF SYSTEM? C PM N ONE**
**S PECTRUM T IMEX 2Ø68**

4. Enter **t** (or T) for TIMEX. The next message will read:

**AERCO Z-DOS V.1.8 DISC NAME?**

5. Type in a name followed by **ENTER.** The disc will stop rotating in a second or two and the formatting is complete. The disc name can be any number of charactors up to **1Ø**, except for " or , (quote or coma). You can now enter **CAT ""**, to view the directory and see the maximum space available on that drive. It is also a good way to verify that an old disc has actually been erased.
6. The next step is to copy the System Boot disc onto the newly formatted disc . If you have a dual drive system;
**6a**. Insert the System Boot disc into **drive B**.
**6b**. Enter the command: **MOVE "a: = b: "**, Read this as: save onto drive **A** the contents of drive **B.** The drive you are copying to is always entered first. Also notice that whenever a drive is designated it is followed by a **colon**. Both drives will first step to the outside. The busy light should alternate between drives as each track is copied. The lights will alternate twice on each track for double sided drives.
7. Reset the computer (switch OFF and ON) and verify that all of the programs on the new disc load properly.

**MOVE** is used to write information onto the disc. The basic command is:

**MOVE "name.extension", decimal parameter(s)**

As an example:
1. Load the adder program: **CAT "adder .bas "**,
2. Select **Y**/yes or **N**/no for a printer to get the main menu.
3. Press **Q**/q to quit followed by **ENTER**.
4. Enter the command: **MOVE "calculator.bas "**,
5. List the directory to see if **calculator .bas** is there.
6. Enter: **CAT "calculator.bas "**, to load the program. You will get the message: **N Statement lost, Ø:255.** Did the system fail? No! Enter **GOTO 1**. The program loaded but did not auto run because no start line was specified in the MOVE or CAT commands.
7. Re-enter the MOVE command with an optional start address:

**MOVE "calculator.bas",1**

8. Re-enter: **CAT "calculator.bas"**, The program will auto run now. Notice there is still only 1 directory listing for calculator.bas.

Loading and saving the other extensions is essentialy the same as for BASIC. You will need to know the **decimal location and length** for binary code. They can often be found in the BASIC listing of a program. There will be a Line that says: **SAVE "NAME" CODE location,length**. An alternate method is to use the **Header** program on the System Boot disc to determine the parameters if the program is entirely in binary code. Once you determine the parameters, enter a MOVE command with the appropriate changes from tape to disc.

## SPECIFICATION OF DISC COMMANDS BY VARIABLES:

Disc commands my be specified from within simple string variables. The first thing to remember is that the BASIC Syntax Checker will always insist on a pair of quotes and a comma to process a Disc Command. For example, MOVE "A$", is the appropriate syntax to save the information specified by A$ to the disc. Now let's consider what needs to be in A$. The general rule is that it will be identical to the equivalent Disc Command typed directly into the system, but there is an important exception. A$ needs to contain a quote after the extension, but it doesn't want to contain a quote before the name. If the direct command would be **MOVE "Name.BIN",1234,5678** then A$ should be equal to **Name.BIN",1234,5678** and it would be saved with **MOVE "A$",** Let's assume that your program has generated A$ equal to Name, the numeric variable Start equal to 1234, and the numeric variable Len equal to 5678. You could then have a program line as follows: **LET A$=A$+".BIN""","+STRING$ Start+","+STRING$ Len** . This will set A$ as required for the move. Note that a quote mark specified within a string statement must be a double quote, as in the double quote after BIN.

## TO INCORPORATE DISC COMMANDS into TAPE PROGRAMS:

The AERCO Centronics print driver tape V2.1 will be used as an example:

1. LOAD the print tape: **LOAD "" ENTER**
2. Exit to BASIC from the menu: Enter **E**
3. Enter: **LIST 14 ENTER**
4. Edit line 14 to read: **PRINT "SAVE ON DISC"**
5. Edit line 23 to read: **"IF C$ ="S" OR C$="s" THEN MOVE "print.bas",999Ø: MOVE "pr.bin",64456,1111**
6. Edit line 9984: **MOVE "print.bas",999Ø: MOVE"pr.bin", 64456,1111**
7. Edit line 9992: **CLEAR 64455. . . : POKE 64459, 79: CAT "pr.bin",**
8. Enter: **GOTO1 ENTER**
9. Enter **S** to save the printer BASIC and CODE onto disc.
10. The contents of the tape can now be loaded from disc with the command:

**CAT"print.bas",**

The directory will show 2 new entries:

**print.bas pr.bin**

If you get the error: **J: Invalid I/O device**, when saving tape programs to disc, enter the command: **OUT 244,1** and re-enter your disc command.

Use the same procedure to modify other programs from tape. LOAD the program from tape. LIST it (if it contains BASIC) and change all of the appropriate commands. Modified programs can be saved onto the disc with an Immediate Command, or the command can be incorporated into the program.

## HOW TO USE ERASE:

To remove individual entries from the disc simply enter the command:

**ERASE "name.extension",**

for instance, to erase the calculator program, enter:

**ERASE "calculator.bas",**

If you want to erase an entire disc, use the **FORMAT** command, but be very careful not to lose something you still want and don't have another copy of. LIST the directory when you are done erasing to verify that the entries are indeed gone.

## SELECTING DRIVES:

With a hard reset (power off then on), the system always auto boots from drive **A**. Unless a different drive is selected within a command, the last drive used remains the selected drive. The 4 drive positions are identified as: **A: B: C: D:** The selected drive is inserted immediately after the open quote and is followed by a **colon**.

**COMMAND " DRIVE: name.extension", decimal parameter**

For example, to LOAD the directory on drive **B:**

**CAT "b:",**

Any subsequent disc commands will now access drive **B**, until a different drive is selected or the system is reset.
If you want to access a different drive from reset, follow this proceedure:

1. Hold down the **Space Bar** when the computer is turned on until the **K** cursor starts skipping across the screen. This defeats the **Auto Boot**.
2. Enter a command with the new drive selected. For example:

**CAT"b:boot.bas"**

The system will boot from drive **B** and continue to operate from that drive until told to do otherwise. It is not neccessary for the boot program to be the first one to run. Any disc command can be entered after the auto boot has been defaulted.

## DISC CAPACITY:

The maximum capacity of a formated disc is dependent on the density (tpi) and number of sides. The following figures are for 3" to 5 1/4" drives.

| | Single Density | Double Density | Quad Density |
|---|---|---|---|
| Single side | 95K | 19ØK | 395K |
| Double side | 19ØK | 395K | 79ØK |

## HOW TO MAKE BACK-UPS:

It is a good idea to get into the habit of making copies of any programs or discs that you can't live without. Much less time is required to copy a disc than to rewrite a program and debug it all over again! An entire disc can be copied on a dual drive system using either the **MOVE** or **CAT** commands (they behave identically in the special case of COPY). Always enter the back-up drive first and the master drive last. To copy **From** drive B: **To** drive A:

**MOVE"A: = B: ", or CAT"A: = B: ",**

To copy an individual file from one disc to another, load the file into system memory first and then save to the second disc. The second disc can be inserted into the same drive or a different drive.

## HOW TO AUTO BOOT YOUR PROGRAMS:

The auto boot feature can be used to load any program you want. On the System Boot Disc, we use it to load the welcome screen (First.SCR). You can use it to dedicate each disc to LOAD a different program on reset. One disc can be used for a Word Proccessor, another for a Spreadsheet, etc. When the system is initialized from reset, it looks for a BASIC program called: **BOOT.BAS** and loads that program. If you have a program written in BASIC or an Assembly language program that includes a BASIC program to

load it (as our Centronics Printer tape), follow this procedure:

1. Load the BASIC program and Assembly code into memory.
2. Make the neccessary changes to incorporate onto disc. See the section:
   **TO INCORPORATE DISC COMMANDS into TAPE PROGRAMS.**
3. Save the BASIC program as **boot.bas**. In the case of the printer program the command would be:

**MOVE"boot.bas",999Ø**

4. The printer program will now auto boot everytime you reset the system with that disc in drive **A.**
5. If your program is in Assembly language only and does not contain a BASIC loader program, you will have to write one yourself.

6. Use a header program to identify the location and length of the program and make the neccessary modifications to disc if it is a tape program.
7. Write a BASIC program to load the machine code. The basic format of the command will be:

```
10 CAT"NAME.bin",location,size
20 RAND USR location
```

## SYSTEM UPGRADES and CHANGES:

The FD-68 Floppy Disc Interface is capable of controlling a maximum of 4 disc drives. Additional drives can be added to any system to bring it to the max. Because the EPROM can be customized for all 4 positions, the drives do not have to all be the same. We can supply custom EPROMs for any combination of drives. Contact us for price.

The 34-conductor signal cable normally supplied with the power supply and drive case combination has provisions to connect 2 drives. There is also a second power connector. Therefore, if you want to add a second drive, the signal and power connections are already provided. If you wish to operate more than 2 drives, contact us for assistance on signal and power connections. (The power pack is capable of operating 3 typical drives.)

We can also provide additional front cover panels for the drive case. Covers are available for single and dual drive combinations (using 2/3 height drives).

## EXPANDING THE ON-BOARD RAM TO 256K:

Since the AERCO FD-68 uses the current industry standard Dynamic RAM components in sockets, the 64K RAM chips can be removed and 256K chips installed in their place. One additional chip and several jumper wires must be installed on the board to accomplish this change, so we don't recommend you do it yourself unless you are **very** experienced in modifying high density circuit boards.

The modification is most easily accomplished if Drives C and D are committed to control the 3 additional 64K pages in the DOCK Bank. This is the most common configuration, giving 2 Floppy Disk drives and 192K Bytes of RAM Disc on top of the 64K Bytes already in the DOCK Bank. (RAM-Disc is a special program that combines the size of storage offered by disk with the speed of access offered by continuous residence in memory). We will so modify your board for $50 plus $3 for ground shipping or $5 for air shipping.

It is also possible, by adding a piggy-back chip on the board, to maintain all 4 floppy disk drives with the expanded RAM. There is an additional charge of $30 for this modification.

## SOURCE CODE AVAILABILITY

We offer commented source code printed out for the convenience of program developers and sophisticated users. It is currently 55 pages and growing, so we charge $20 for printing it. As soon as the CP/M system disk is available, we will offer machine-readable source code for the same price. Note that a full set of schematics is included in this manual. We will cheerfully answer all questions concerning the detailed operation of the system and provide as much individual support as we can practically afford. Our intention is to provide a completely general-purpose open-architecture interface useful to users, developers, and manufacturers who wish to base products on the TS-2068.

## TROUBLESHOOTING PROBLEMS:

All systems and components are tested before they are shipped. Be sure to follow directions when you assemble the system. **NEVER** add or remove components when power is **ON.** If you continue to have difficulties, we are available for direct phone assisstance **(512-451-5874)** or by mail **(P.O.Box 18Ø93, Austin, TX 7876Ø)**. If it becomes neccessary to return equipment for repair, it is best to ship by **UPS**. Address all shipments to: **AERCO, 76Ø6 Robalo Rd., Austin, TX 78757**.
If you need assistance with programming techniques, send inquiries to our P.O. Box. Try to include as much information as possible. A copy of the program tape and/or disc is usually helpful. We will try to provide as much support as possible, but other commitments often over-ride one-of-a-kind applications. Additional support is also available from the USR Groups and T/S publications.

The following is a summary of the most common sources of problems. Be sure that system components are compatible. The proper **EPROM** must be supplied for the drives used. The system will not run if the **Track to Track Access time** of the EPROM is faster than the drive. The **System Boot Disc** must be appropriate for your drive(s) (48 / 96 TPI).

**FD-68 INTERFACE:** Check the connection between the board and the computer as well as accessories on the feed-thru. Be sure the 34-conductor signal cable and RGB cables are properly attached. Check to see that all of the IC chips in sockets are fully inserted. Visualy inspect the board and the connectors for obvious defects such as cracks or foreign objects.

**DISC DRIVES:** Check signal and power cable connections. Insure that Drive Select and Terminating Resistor placement is correct. If there is only 1 drive, it must have the resistor. If there is more than 1 drive, **ONLY 1** of the drives should have it. Ensure that the System Boot Disc is properly inserted. The elongated read/write slot always goes to the rear. The Write Protect Tab goes toward the red busy lite on some brands and away on others. It will not hurt to experiment, inserting it both ways if you are not sure. The system will report the Disk Not Rotating if it is inverted.

**J: Invalid I/O device** error code means you must enter the following command: **OUT 244,1** before the disc can be accessed. This is due to errors and inconsistencies in the Timex ROM Bank Switching code. As these errors are found and corrected, smoother system operation will eventually be realized. (Remember that the fully populated DOCK Bank in the FD-68 interface means that the Timex "operating system" is running from RAM instead of ROM, so modifications and corrections are accomplished without physically changing the ROM.)

**N statement lost** when loading a BASIC program means you didn't give an optional start address. **GOTO 1** (or the start of the program) will run it.

If your system will not auto boot, try the following steps:
1. Try to read the directory: **CAT"",**
2. If the directory loads, enter: **CAT"boot.bas",**

If you have a second drive, default the auto boot and try drive B.
1. Reset the computer and hold the **Space Bar** down until the **K** cursor skips across the screen.
2. Try to boot drive B: **CAT"B:boot.bas",**

We have had boot diskettes damaged during shipping, presumably from the large magnets in the x-ray parcel inspection systems. We are now ensuring that the boot disk is packed in a carton with extra space around it to isolate it from magnetic fields. If your system won't read the boot disk, try to isolate the problem to the FD-68 system or the boot disk with the following procedure:

1. Insert a fresh disc into the drive .
2. Enter: **FORMAT"",**

If the system works, it will print **FORMATTING** with a * for each track. When it is done, enter **t/T** for Timex system and a name followed by **ENTER**.
3. You can now view the directory with: **CAT"",**

If you can get this far, your system is reading and writing diskettes. Try entering a short BASIC program from the keyboard and save it with
**MOVE "Boot.BAS",1** Turn the power off and then back on. See if the system auto-boots your program. Contact us for a replacement boot disk or additional assistance as required.

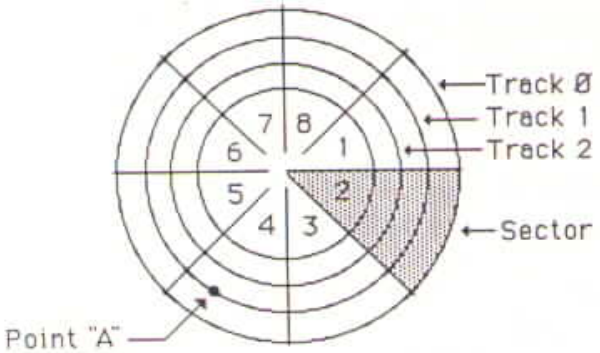
All of the drives we sell are double sided drives. That means that they write on both sides of the diskette. The system automatically starts writing on the second side of the disk when the first side is full. You would never be expected to turn the disk over. Don't use "flippy" disks that are meant to be turned over in single-side drives. If you were to insert such a disk inverted from the way it was formatted you would confuse the directory and surely lose files.

There is a wide range of prices charged for diskettes. It is certainly not necessary to buy the most expensive ones, but neither is it prudent to save valuable files on the cheapest ones. In general, single side diskettes are **NOT** reliable on the second side. We are told that all diskettes are created equal and then tested and marked accordingly. If a diskette has two fully functional sides, it will be marketed as a double side disk.

# DISC OPERATION THEORY:

A disc can most easily be compared to a phonograph record. Data files are recorded in binary form and stored in the concentric grooves (tracks) of the disc. They are accessed by means of a read/write head which is partially analogous to the tone-arm of a record player. However, it is not necessary to move around each track in sequence to reach a particular point; the desired location can be arrived at directly (directly or random-access) by means of the track/sector method of addressing.

For addressing purposes, discs are formatted into tracks and sectors. Tracks are numbered sequentially, starting with the outermost track and working inward. Sectors are wedge-shaped and are numbered around the disc. (Note: sectors are electronic divisions only; they are not visually discernable.)




Example: a program starting a Point "A" would be located at track 1, sector 4.

The illustration used is a simplification. A typical disc has 35/4Ø/8Ø trtacks per side with 1Ø sectors each.

On most floppy discs, tracks and sectors are located by utilizing a series of magnetic markers or **headers** which are spaced around the tracks at the beginning of each sector. The disc is kept constantly spinning at high speed (36Ø RPM); as the read/write head passes over each header, magnetically-encoded information including track and sector number, which side of the disc, and size of the track sector (256 bytes SD/512 bytes DD) is relayed to the disc drive circuitry ( all at the rate of 25ØK bits/sec). Thus, to access a particular address, the drive moves the head to the correct track and then locates the desired header.

An additional means of sector orientation is the **index hole** in the disc, located near the center, marking the beginning of sector 1. As the disc spins, the hole periodically aligns with another hole in the disc envelope, allowing a ray of light to pass through to a sensor in the drive circuitry. Although this does allow calibration from sector 1, its main purpose is to provide a check on the operation of the disc itself. The disc circuitry will be alerted to any irregularity of the disc spin ( slowing, stopping, etc.) by changes in the timing of the sensor ray.

# FD-68 THEORY of OPERATION:

Although it is beyond the scope of these instructions to provide a detailed theory of the interface, this summary is provided. Some of the essential components and their function are as follows. Refer to the schematics and assembly drawing for details:

**FD1797 Disc Controller:** Interfaces the 8-bit data bus on the Timex to a serial bus for the disc drive. Controls the direction and side select (for double sided drives) of the read/write head operation. Data read from the disc is processed at the industry standard rate of **25ØK bits/sec.**

**9216-B:** Is a high speed Data Separator used to "condition the sequence" of data as it comes off the disc into the Disc Controler.

**2764 EPROM:** Stores the **D**isc **O**perating **S**ystem (DOS). This information is loaded into 8K of the 64K of additional RAM when the system is initialized. Information is also stored on the size, density, sides and speed of all four drive positions.

**4764 DYNAMIC RAM:** Provide the additional 64K of memory consisting of **8** each **8K chunks**. Located in the Dock Bank and accessed thru the Bank Switching facility of the Timex. When configured as a Timex system, the Timex ROM (with modifications) is stored in **Chunk Ø** and the FD-68 ROM is loaded into **Chunk 1**. When configured as a CP/M system, all **8** Chunks will be available for program and data. Can also be modified for **256K**.

**74LS157:** Two Multiplexer chips are used to decode the 16 bit address bus of the computer. They select the appropriate address in RAM to perform each function at the correct time and sequence.

**74LS245:** Is a Buffer for the **data bus**. It provides a two-way interface beteen the computer and the Disc Controller, RAM,ROM and Latch.

**74LS273:** Is a latch used to decode instructions to select the disc drives and turn on the motors.

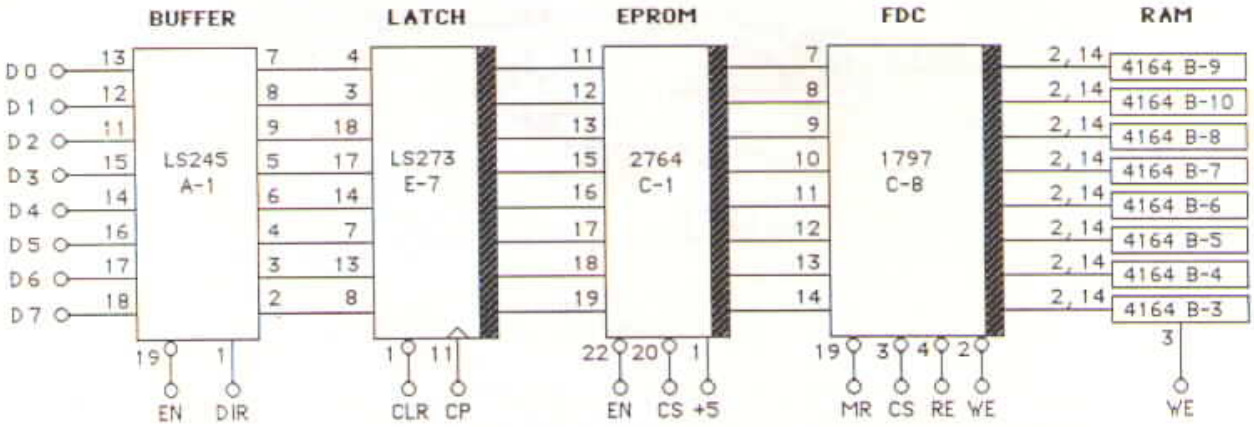
**74HØ4 (74Ø6):** Are open collector inverters used to control the various electro/mechanical functions of the drives. They are capable of handling relatively high currents.

The **crystal, 74LSØ4 (A-8)**, and **74LS393** are used to generate the clock signals used by the Disc Controler **(1 MHZ/ 2 MHZ)** and Data Seperator **(8 MHZ)**.

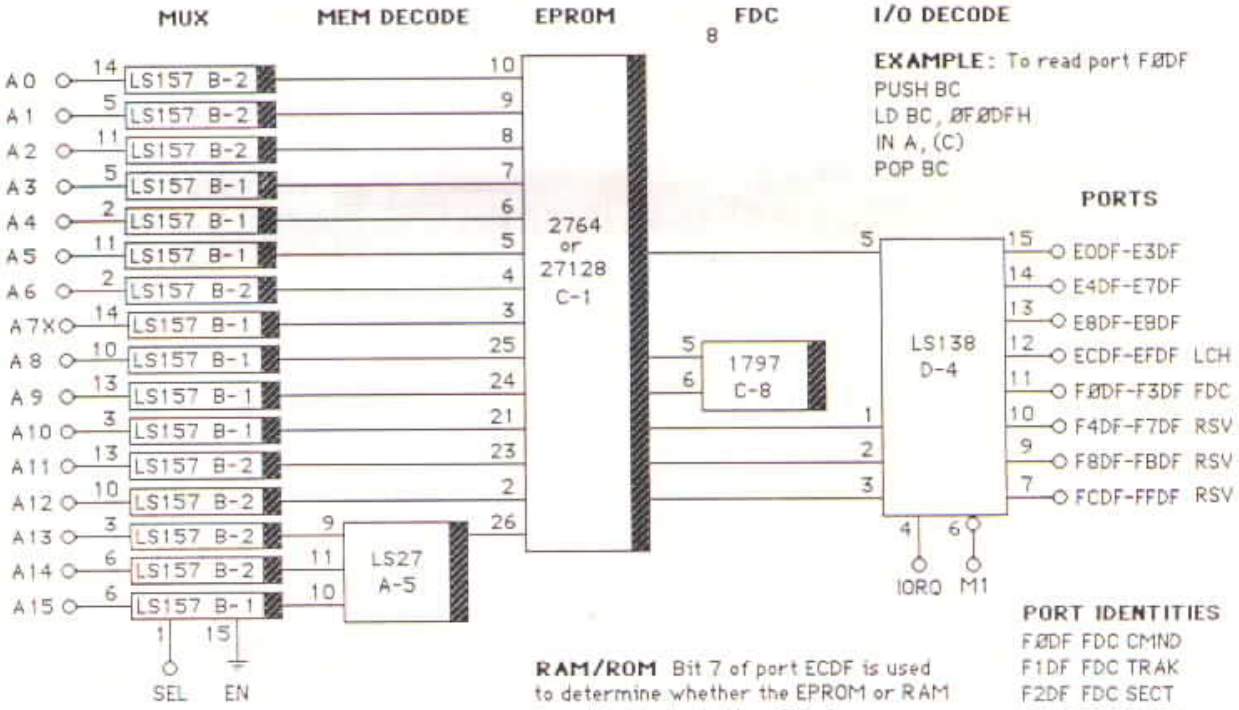
AERCO
ACME ELECTRIC ROBOT CO.

# FD-68 FLOPPY DISC INTERFACE © 84 AERCO

## DATA BUS




## ADDRESS BUS




**EXAMPLE:** To read port F0DF

```
PUSH BC
LD BC, 0F0DFH
IN A, (C)
POP BC
```

**RAM/ROM** Bit 7 of port ECDF is used to determine whether the EPROM or RAM is enabled in the bottom 8K of memory. Bit 7 is set LOW on Reset, EPROM selected.

**DO NOT** read ports ECDF-EFDF. This will set all bits of the latch high.

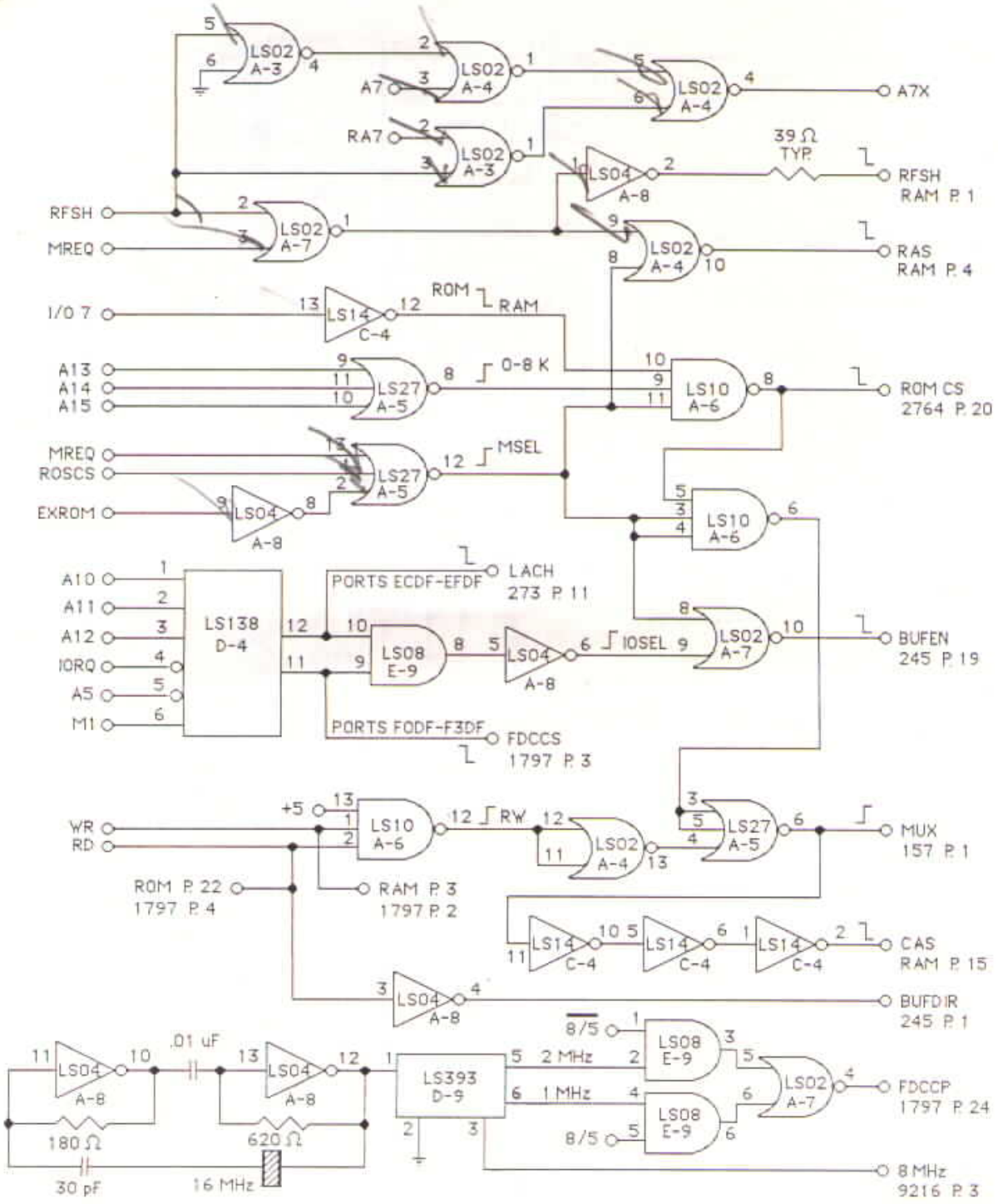
**PORT IDENTITIES**

F0DF FDC CMND
F1DF FDC TRAK
F2DF FDC SECT
F3DF FDC DATA
ECDF LATCH
E8DF USED BY RS 232
E4DF USED BY RS 232
E0DF USED BY CENT.

AERCO
ACME ELECTRIC ROBOT CO.

# FD-68 FLOPPY DISC INTERFACE © 84 AERCO

## CONTROL LOGIC

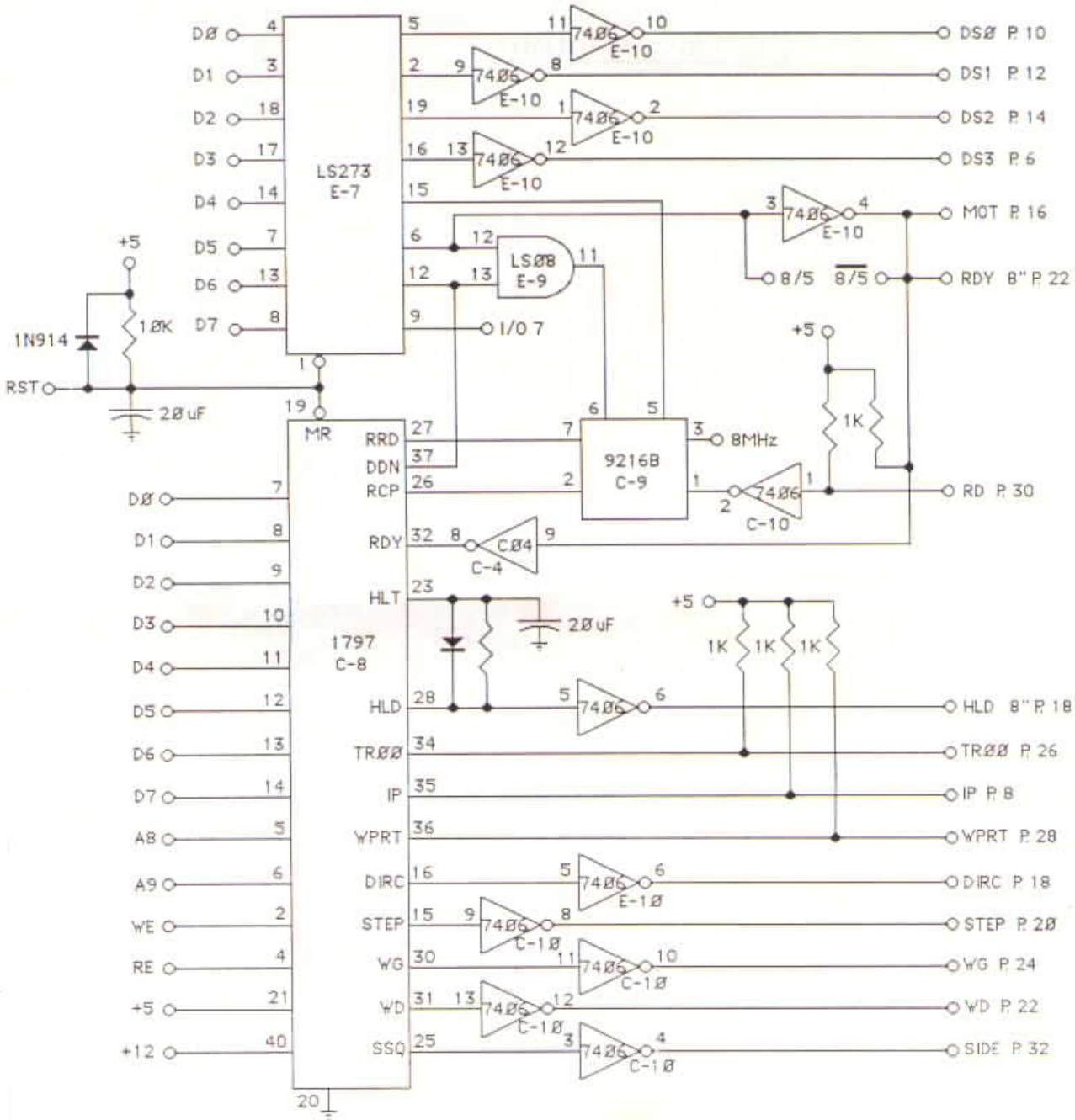


AERCO

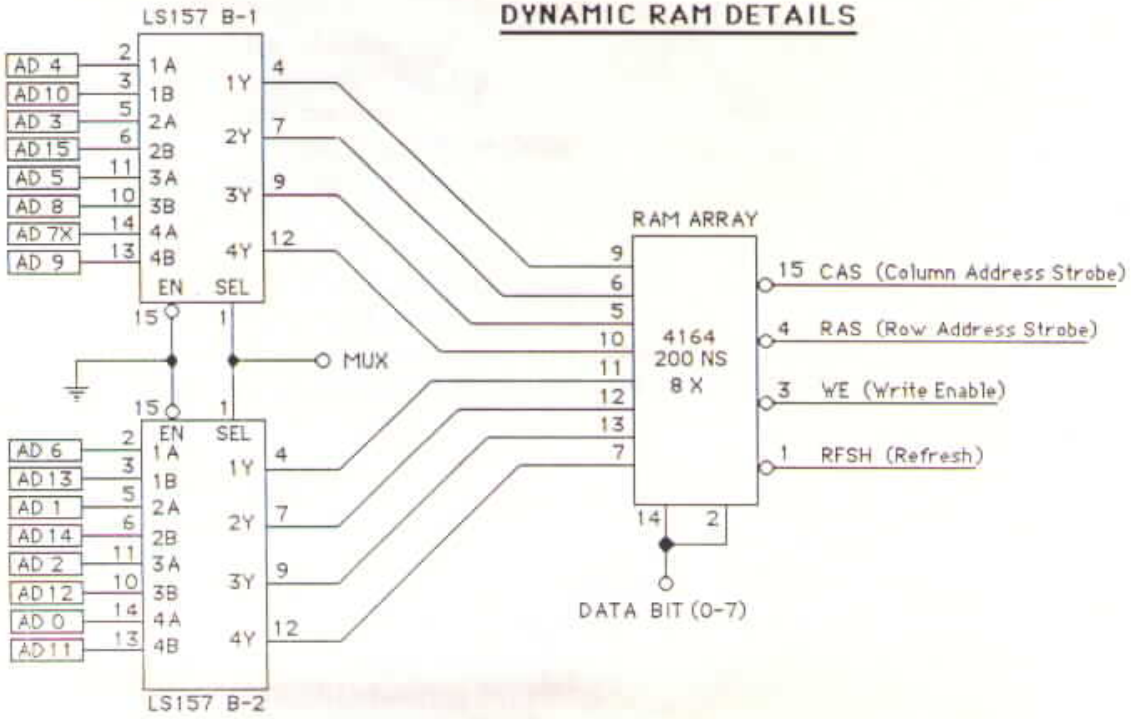
ACME ELECTRIC ROBOT CO.

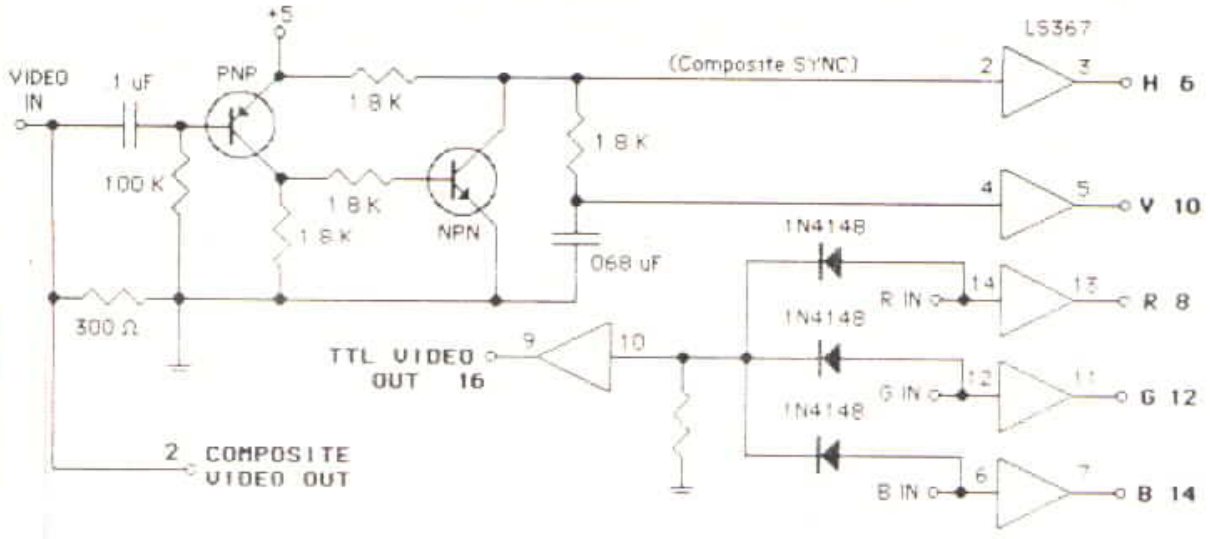
# FD-68 FLOPPY DISC INTERFACE © 84 AERCO

## DISC CONTROLLER



## DYNAMIC RAM DETAILS




## RGB VIDEO DETAILS